

Una `TabTreni` è una struttura dati (implementata a lista) che mantiene informazioni sui treni in partenza da una stazione ferroviaria. Per ciascun treno in partenza, la struttura dati deve poter visualizzare l'ora di partenza, il minuto di partenza, e la stazione di destinazione. Tutti i nomi delle stazioni sono al più di 30 caratteri. Le operazioni che possono essere fatte su una `TabTreni` sono le seguenti:

✓ `inizializzaTab(T, src);` [3pt]

Funzione che inizializza la struttura dati `T` per la stazione ferroviaria di partenza `src`. Inizialmente non ci sono treni in partenza.

✓ `nuovoTreno(T, h, m, dest);` [5pt]

Funzione booleana che aggiunge un treno in partenza per la stazione di destinazione `dest` alle ore `h:m` (`h` è un intero che va da 0 a 23, `m` un intero da 0 a 59). Il treno viene inserito nella struttura dati in ordine temporale di partenza. Non possono esistere nella struttura dati due treni con lo stesso orario `h:m` di partenza. La funzione restituisce `true` se l'inserimento ha successo, `false` altrimenti.

✓ `cancellaTreno(T, h, m, dest);` [5pt]

Funzione booleana che cancella il treno in partenza all'orario `h:m` per la destinazione `dest`. Se l'eliminazione ha successo, la funzione restituisce `true`, altrimenti restituisce `false`.

✓ `partenzaTreno(T);` [5pt]

Funzione booleana che elimina dalla struttura dati `T` il treno con partenza più imminente, qualora sia presente almeno un treno. Se l'eliminazione ha successo, la funzione restituisce `true`, altrimenti restituisce `false`.

✓ `stampaTabTreni(T);` [4pt]

Funzione che stampa a video la struttura dati `T`, secondo il seguente formato:

```
Treni in partenza da PISA
[10:30]TORINO
[10:45]MILANO
```

In questo esempio ci sono solo due treni in partenza dalla stazione di PISA. I treni sono elencati in ordine di partenza. Qualora non ci fossero treni la funzione solo "Treni in Partenza da PISA" e basta.

✓ `quantiTreni(T);` [3pt]

Funzione che restituisce il numero di treni in partenza.

Mediante il linguaggio C++, implementare il tipo `TabTreni` definito dalle precedenti specifiche utilizzando le **strutture**. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

Esempio di funzione main()

```
int main( ){

    TabTreni T;
    inizializzaTab(T, "PISA");

    nuovoTreno(T, 10, 30, "TORINO");
    nuovoTreno(T, 10, 45, "MILANO");
    nuovoTreno(T, 9, 30, "TORINO");
    nuovoTreno(T, 10, 45, "ROMA");           //fallisce

    stampaTabtreni(T);                       // Treni in partenza da PISA
                                           // [9:30] TORINO
                                           // [10:30] TORINO
                                           // [10:45] MILANO

    int m = quantiTreni(T);                  // m=3

    cancellaTreno(T, 10, 30, "TORINO");

    stampaTabtreni(T);                       // Treni in partenza da PISA
                                           // [9:30] TORINO
                                           // [10:45] MILANO

    partenzaTreno(T);
    stampaTabtreni(T);                       // Treni in partenza da PISA
                                           // [10:45] MILANO

    return 0;
}
```

Domande

1. **[2pt]** Rappresentare il numero intero -12 in complemento a 2 su 8 bit, se rappresentabile.
2. **[1pt]** Scrivere una funzione che dati due array V e W di interi di dimensione dim , restituisce `true` se ogni elemento dell'array V è presente nell'array W , `false` altrimenti.
3. **[2pt]** Scrivere una funzione **ricorsiva** che prende in ingresso un numero intero positivo e restituisce la somma del quadrato dei primi n numeri.
Esempio: $f(3) = 0^2 + 1^2 + 2^2 + 3^2 = 14$.